



Rigorous engineering of collective adaptive systems – 2nd special section

Martin Wirsing¹ · Stefan Jähnichen^{2,3} · Rocco De Nicola⁴

Accepted: 2 November 2023 / Published online: 14 November 2023
© The Author(s) 2023, corrected publication 2023

Abstract

An adaptive system is able to adapt at runtime to dynamically changing environments and to new requirements. Adaptive systems can be single adaptive entities or collective ones that consist of several collaborating entities. Rigorous engineering requires appropriate methods and tools that help guaranteeing that an adaptive system lives up to its intended purpose. This paper introduces the special section on “Rigorous Engineering of Collective Adaptive Systems.” It presents the 11 contributions of the section categorizing them into five distinct research lines: correctness by design and synthesis, computing with bio-inspired communication, new system models, machine learning, and programming and analyzing ensembles.

Keywords Adaptive systems · Collective adaptive systems · Software engineering · Formal methods · Rigorous methods

1 Introduction

A collective adaptive system (CAS), often referred to as an ensemble, is composed of collaborating entities capable of adjusting dynamically to evolving and open-ended environments, as well as to new requirements that arise during runtime [46, 51]. Often, the individual entities within such a system possess distinct properties and objectives. The interactions among these entities, or with humans, can lead to unforeseen emergent phenomena.

In the contemporary landscape, numerous systems exemplify both collectiveness and adaptiveness. Notable examples include intelligent systems such as smart cities and advanced transportation networks, cyber-physical systems like swarms of robots and sensor networks, and socio-technical systems like distributed energy grids.

The rigorous engineering of CASs demands the application of appropriate methodologies and tools to ensure these systems maintain alignment with their intended purposes even in the face of dynamically changing environments. This comprehensive endeavor encompasses:

- Theory development: Creating robust theories for the design and analysis of CASs.
- Programming and management techniques: Developing specialized techniques for effectively programming and managing these intricate systems.
- Systematic mechanism creation: Employing systematic methods for crafting adaptive mechanisms that respond intelligently to evolving conditions.
- Validation and verification: Implementing rigorous techniques for validating and verifying the functionality and reliability of these systems.
- Security and trust enhancement: Incorporating strategies to enhance security and trustworthiness, ensuring the integrity of the system’s operation.
- Performance optimization: Striving for performance optimization and enhancing efficiency while preserving system adaptability and effectiveness.

✉ M. Wirsing
wirsing@lmu.de

S. Jähnichen
stefan.jaehnichen@tu-berlin.de

R. De Nicola
rocco.denicola@imtlucca.it

¹ Ludwig-Maximilians-Universität München, München, Germany

² TU Berlin, Berlin, Germany

³ FZI Forschungszentrum Informatik, Berlin, Germany

⁴ IMT School for Advanced Studies Lucca, Lucca, Italy

Research projects and workshop series The European Commission was one of the main drivers of the research on ensembles and CASs.

The starting point was the “Global Computing Initiative”¹ with projects such as AGILE [5] and MIKADO [12] on mobile computing [64] and with the Interlink workshop series [71]. Subsequently the influential European research programs “Self-Awareness in Autonomic Systems”² and “Fundamentals of Collective Adaptive Systems”³ funded projects such as ASCENS [72, 74] on “Autonomic Service-Component ENsembles,” SAPERE [77] on “Self-Aware PERvasive service Ecosystems,” ALLOW Ensembles [24] on large-scale collective adaptive ensembles, and QUANTICOL [18] on a quantitative approach to the design and management of collective and adaptive behaviors.

More recent related initiatives and projects are the German Transregional Collaborative Research Centre 248 “Foundations of Perspicuous Software Systems”⁴ and the Italian collaborative project “IT-Matters”.⁵ There are several related workshop series, e.g., on “Engineering Collective Adaptive Systems”⁶ and on “Formal Methods for Autonomous Systems (FMAS)”⁷.

This special section on “Rigorous Engineering of Collective Adaptive Systems” is a follow-up of the first special section on this topic [35] and was inspired by five successful tracks [34, 50, 73, 75, 76] at the ISoLA conferences in 2014 [54], 2016 [55], 2018 [56], 2020 [57], and 2022 [58].

Contents The 11 papers included in this special section are revised and extended versions of papers originally presented at the ISoLA 2022 track [76]. They all closely align with the global challenges outlined above ranging from advancing theoretical foundations to ensuring correctness and optimal performance. In the following sections, we provide brief overviews of the papers, categorizing them into five distinct research lines:

- correctness by design and synthesis,
- computing with bio-inspired communication,
- new system models,
- machine learning,
- programming and analyzing ensembles.

¹ <https://cordis.europa.eu/programme/id/FP5-IST-2001-6.2.2>, last accessed 2023-08-16.

² https://cordis.europa.eu/programme/id/FP7_ICT-2009.8.5, last accessed 2023-08-16.

³ https://cordis.europa.eu/programme/id/FP7_ICT-2011.9.10, last accessed 2023-08-16.

⁴ <https://www.perspicuous-computing.science/>, last accessed 2023-08-16.

⁵ <http://itmatters.imtlucca.it/>, last accessed 2023-08-16.

⁶ See, e.g., the 6th eCAS Workshop on Engineering Collective Adaptive Systems, <https://apice.unibo.it/xwiki/bin/view/ECAS2021/>, last accessed 2023-08-16.

⁷ <https://dblp.dagstuhl.de/db/conf/fmas/index.html>, last accessed 2023-08-16.

For each category, we also highlight recent research results related to the respective line.

2 Correctness by design and synthesis

Building correct software systems presents a grand challenge. Although CASs often exhibit considerable complexity, certain critical components can be synthesized or constructed correctly through compositional reasoning processes. The two papers in this section exemplify two complementary approaches.

The paper titled “Correct by Design Coordination of Autonomous Driving Systems” [22] by Marius Bozga and Joseph Sifakis presents a “correct-by-design” approach for ensuring accurate coordination in autonomous driving systems. These systems are real-time distributed systems where components possess only partial knowledge of their environment. Their behavior is subject to influence from temporal and spatial changes, and they must simultaneously satisfy both individual and collective goals. The development of reliable autonomous driving systems entails addressing significant challenges, including system modeling, specification of traffic rules, system construction, and verification of system correctness.

The approach of Bozga and Sifakis builds upon earlier work [21] related to collision avoidance policies and the representation of autonomous driving systems, combining static environment maps with vehicle dynamics. An autonomous driving system is seen as a dynamic system consisting of a map and a set of vehicles moving along specific routes. It is managed by a runtime which computes the available spaces for vehicles based on their positions and attributes. Applying assume-guarantee reasoning, Bozga and Sifakis establish the principle of a safe control policy for vehicles, ensuring they adhere to their allocated free space. They introduce speed control policies that both comply with safety contracts and are non-blocking. Traffic regulations are expressed through linear-time temporal logic formulas that impose speed limitations on vehicles. The primary outcome is the development of secure free space policies, guaranteeing that the system’s behavior, when implemented, is inherently safe in accordance with the contracts governing the vehicles and the runtime.

Strategy synthesis in games is a timely research field (see [45] for an overview) with applications in many domains, encompassing transportation, medical systems, and CASs. Integrating strategy synthesis with methods like spatial model checking is a promising topic which can lead to better modeling and analysis capabilities for CASs.

In the second paper of this section, titled “An Experimental Toolchain for Strategy Synthesis with Spatial Proper-

ties” [67], by Maurice ter Beek, Davide Basile, Laura Bussi, and Vincenzo Ciancia, the authors demonstrate the practical application of strategy synthesis to ensure spatial properties. They introduce a toolchain that seamlessly integrates CATLib [6] and VoxLogicA [8], enabling full automation. CATLib is a tool for game composition and strategy synthesis relying on finite state automata, while VoxLogicA is a spatial model checker facilitating the verification of properties related to (pixels of) digital images. The authors exemplify their approach with two case studies in strategy synthesis on automata: one is a fundamental example that encodes agent motion in image-based spaces, and the other is a realistic proof-of-concept derived from a railway domain case study. Notably, the latter example is a novel addition, and substantial enhancements have been made to both the toolchain and the model-checking procedure compared to the paper presented at the ISOLA 2022 conference [7].

3 Computing with bio-inspired communication

Biological and natural systems, like bird flocks, ant colonies, and insect swarms, have captivated researchers due to their complex, emergent behaviors. Understanding these phenomena has involved diverse mathematical approaches, including graph theory [62], statistical mechanics [17], and optimization algorithms [40].

The paper “A causal, time independent synchronization pattern for collective adaptive systems” by Peter Fettke and Wolfgang Reisig, a revised version of their previous work [42], relies on Petri nets for formal foundations and advocates the Heraklit approach [41] for modeling the behavior of artificial ants. Heraklit is a modern component-based modeling language and tool. Dynamic behavior is expressed by symbolic Petri nets and thus structured along causal dependencies, and not along temporal relations. The Heraklit approach is exemplified through a case study involving “ants on a bar” and the “net-atom” design pattern is introduced as a generalization of the ant example.

Heraklit, as well as most of the abovementioned approaches, focuses on global, top-down system behavior. In contrast, the paper “Modelling Flocks of Birds and Colonies of Ants from the Bottom Up” [38] by Rocco De Nicola, Luca Di Stefano, Omar Inverso, and Serenella Valiani argues for a different, bottom-up approach. This approach involves defining systems in terms of their individual components and local rules, allowing for the natural emergence of collective behavior. The advantages of this approach are demonstrated through a diverse set of examples, including instances of flocking behavior, ant colonies, and group dynamics. The paper places a strong emphasis on the use of formal languages to facilitate the use of automatic tools for efficient

simulation and formal analysis. This approach paves the way for scalable, adaptive system designs that draw inspiration from principles found in natural systems. Building upon the work presented in [37], this paper extends the research by introducing fresh examples and presenting new evaluation results to substantiate and further validate the proposed approach.

4 New system models

Several approaches to system modeling in the context of CASs are known. These approaches encompass various methodologies, including process-oriented system models, exemplified by the SCEL language [33], the ensemble system model [47], and a stream-based approach [23]. These system models consider collections of individual entities as ensembles. When dealing with a large number of agents, individually examining each agent’s state becomes impractical. For example, in scenarios involving swarms of droids or autonomous taxis, tracking the details of each agent is prohibitively costly or even unfeasible. Instead, the focus shifts toward achieving the collective objectives of the group, such as coordinating droid choreography or ensuring uniform taxi coverage in urban areas. Statistical approaches such as chemical reaction networks (see, e.g., [27]), fluid approximation (see, e.g., [19]), or mean-field approximation (see, e.g., [20]) are preferred for analyzing the collective behavior of such systems. These approaches treat the agents’ states as real numbers and concentrate on aggregate properties, offering a more practical means of studying large-scale CAS scenarios.

The paper “A Kinetic Approach to Study the Collective Dynamics of Multi-Agent Systems” [60] by Stefania Monica, Federico Bergenti, and Franco Zambonelli introduces the Kinetic Theory of Multi-Agent Systems (KTMAS), which is a statistical approach for analyzing the asymptotic collective dynamics of decentralized multi-agent systems. KTMAS is specifically designed to investigate the long-term behavior of these systems, where agents influence each other’s states through direct messaging. The paper builds on previous work [59], extending it to provide an analytic framework for characterizing KTMAS. This framework supports both descriptive and prescriptive reasoning for understanding and designing the long-term behavior of large multi-agent systems. Importantly, it offers an alternative to simulations. By applying the framework to multi-agent systems employing the symmetric gossip algorithm, the authors demonstrate the accuracy and effectiveness of KTMAS. They provide rigorous proof for two asymptotic properties of the symmetric gossip algorithm and validate them by comparing the results with independent simulations conducted on a multi-agent implementation.

The second paper in this section “A Modal Approach to Conscious Social Agents” [30] by Chen Yifeng and J. W. Sanders aims at a fundamental mathematical model for the notions of awareness, consciousness, and empathy of agents. The quest for a universally accepted definition of consciousness has been elusive across diverse fields like neuroscience, philosophy, physics, computer science, and mathematics. In response, this paper takes an innovative approach by introducing laws of modal logic that are designed to describe and capture the essential properties of consciousness. The ultimate goal is to develop a comprehensive set of laws that characterize consciousness, offering a fresh perspective on this complex and multifaceted phenomenon. The study builds upon the approach introduced in prior work by the same authors [29] and focuses on systems where agents coexist within a habitat alongside an external environment. This ecosystem undergoes dynamic changes over time, involving shifts in agent behaviors, interactions, and adaptations. The research investigates agent interactions and empathy with a particular emphasis on awareness, utilizing reflexive theories to infer properties of consciousness, specifically as they relate to awareness of awareness. The theory is founded on falsifiable observations, with laws expressed based on this framework, leading to a collection of consistent laws. The study also delves into empathy, especially among eusocial animals (colonial animal species that live in multigenerational family groups) and creatures with social interactions, and applies this formalism to human empathy tests and AI-related scenarios – the so-called Sally–Anne Test of human empathy and the Keynesian Beauty Contest.

5 Machine learning

Many modern smart systems, like Industry 4.0 manufacturing systems, are self-adaptive systems, which continually monitor their performance, adapt their behavior, and optimize themselves to align with changing conditions. Such systems often comprise multiple components that need coordinated adaptation and optimization. Traditionally, component-based frameworks and adaptation rules (see, e.g., [25, 36, 65]) were used for guiding system reconfiguration, but as data-driven approaches gain prominence, machine learning and neural networks are used for controlling how cooperating agent systems are dynamically formed and reconfigured at runtime (see, e.g., [9, 44, 53, 70]). Neural networks are integrated into the self-adaptation loop, specifically the MAPE-K loop [48], which guides real-time decisions and architectural changes, such as service routing and deployment. Traditionally, component-based frameworks and adaptation rules (see, e.g., [25, 36, 65]) were used for guiding system reconfiguration. However, with the

increasing prominence of data-driven approaches, the landscape has evolved. Machine learning and neural networks now play a pivotal role in orchestrating the dynamic formation and reconfiguration of cooperating agent systems at runtime (see, e.g., [9, 44, 53, 70]). Neural networks have been seamlessly integrated into the self-adaptation loop, particularly within the MAPE-K loop framework [48]. This integration guides real-time decisions and architectural changes, encompassing tasks such as service routing and deployment.

Nevertheless, a noteworthy challenge in this domain concerns the somewhat ad-hoc integration, often lacking a well-defined architectural framework. The core of this challenge lies in the absence of abstract architectural models capable of elegantly encapsulating machine learning and data-driven optimization strategies across many different system components.

The paper “Machine-learning abstractions for component-based self-optimizing systems” authored by Michal Töpfer, Milad Abdullah, Tomáš Bureš, Petr Hnetynka, and Martin Kruliš effectively addresses this gap by introducing a novel component model called ML-DEECo. This model features dedicated abstractions for supervised learning and optimization heuristics, with a demonstrated use case in Industry 4.0 settings. The authors provide a Python-based implementation that illuminates the entire automated learning process, from data collection to real-time inference. This not only streamlines the development of the learning loop but also reduces the level of expertise required for implementation. The paper provides also a detailed description of how machine learning is integrated into the ML-DEECo component model.

Neural networks also offer the advantage of learning unforeseen stochastic relationships (see, e.g., [9, 39]), making adaptations more refined and dynamic. However, switching entirely from rule-based to neural network-based adaptations can lead to the loss of domain knowledge and complexity issues.

The paper “Generating Adaptation Rule-Specific Neural Network” authored by Tomáš Bureš, Petr Hnetynka, Martin Kruliš, František Plášil, Danylo Khalyeyev, Sebastian Hahner, Stephan Seifermann, Maximilian Walter, and Robert Heinrich addresses two crucial research questions: how to infuse domain knowledge encoded in logical rules into neural network learning and how to incrementally transition from rules to neural networks. To tackle these issues, the authors introduce the rule-specific neural network (rsNN) method. This approach systematically converts adaptation rules into corresponding rsNNs, tailored to the logical predicates found within the rules. Additionally, atomic “attunable” predicates further refine these constructs before integration. The paper includes motivating examples and illustrates the rsNN approach.

6 Programming and analyzing ensembles

The last part of this special section discusses novel programming and analysis techniques for multi-robot systems and reconfigurable interacting systems.

Developing software for a single autonomous robot is challenging, but the complexity amplifies in multi-robot systems (MRSs), where seamless cooperation among robots is imperative. Existing software tools like Robot Operating System (ROS; see, e.g., [52]) have undoubtedly facilitated robotics application development. However, they often necessitate grappling with low-level intricacies, which can be time consuming and error-prone. To alleviate these challenges, researchers have proposed the adoption of model-driven engineering principles and higher-level language abstractions to streamline development processes (see, e.g., [28]). In particular, in [14], an approach was introduced integrating the X-KLAIM language [13] with ROS for single robots, and this approach was later extended to MRSs [15]. X-KLAIM [11] is a programmer-friendly extension of the coordination language Kernel Language for Agents Interaction and Mobility (KLAIM; [32]) with high-level process constructs. It is worth noting that KLAIM served as the foundational inspiration for developing the kernel language Service Component Ensemble Language (SCEL; [33]), designed specifically for crafting CASs.

The paper “Coordinating and Programming Multiple ROS-based Robots with X-KLAIM” [16] authored by Lorenzo Bettini, Khalid Bourr, Rosario Pugliese, and Francesco Tiezzi represents a significant enhancement to the approach initially introduced in [15]. In this paper, the authors introduce a novel structuring of robot behavior into distinct layers, resulting in more readable and maintainable code. Two warehouse scenarios showcase the incremental development enabled by this methodology for MRS. The proposed framework serves as a proof-of-concept for applying the tuple space-based paradigm to the realm of MRS software development. With experiments comparing a Java-based implementation against the conventional ROS Python-based approach, the paper demonstrates slightly higher but acceptable latency and lower memory consumption in the Java code, underlining the practicality of their approach.

Like ensembles, Reconfigurable Interacting Systems [1], or Reconfigurable MASs for short, consist of a set of autonomous agents that interact based on mutual interest. These systems exhibit opportunistic interactions through reconfiguration, group formation, and self-organization. Typically, Reconfigurable MASs deal with smaller-scale systems than CASs.

The paper “Language Support for Verifying Reconfigurable Interacting Systems” by Yehia Abd Alrahman, Shaun Azzopardi, Luca Di Stefano, and Nir Piterman presents R-CHECK, a toolkit for designing and verifying Reconfigurable MASs. R-CHECK builds upon ReCiPe’s symbolic

semantics and extends it with a minimalistic high-level programming language. Furthermore, it integrates the interaction logic Itol [1] to facilitate reasoning about interaction strategies. R-CHECK offers a comprehensive toolkit that includes simulation and visualization of model checking counterexamples. This functionality empowers users to perform thorough verification of high-level features, such as channel mobility, reconfiguration, coalition formation, and self-organization. The initial version of R-CHECK was introduced in [2]. The paper in this issue, referenced as [3], extends R-CHECK’s capabilities. It now includes native support for reasoning about message exchange and incorporates a native interpreter designed specifically for the semantics of R-CHECK models.

The paper “Comparing Performance Abstractions for Collective Adaptive Systems” [61] by Maurizio Murgia, Riccardo Pincioli, Catia Trubiani, and Emilio Tuosto focuses on the quantitative analysis of CASs. Utilizing a multi-robot scenario, the paper demonstrates the application of two distinct approaches for performance analysis: one based on queuing networks [49] and the other on generalized stochastic Petri nets [31, 63] (for another use of Petri nets in CASs, see [43] in this issue). These approaches differ in their methodologies. The former automatically compiles the model from a behavioral specification in the style of the AbC calculus [4], while the latter requires the designer to directly create a performance model. The paper investigates three different coordination scenarios: independent non-communicating autonomous robots, collaborating robots, and a centralized architecture where the robots are guided by a coordinator. The paper extensively compares and discusses the advantages and disadvantages of the various specification methodologies and architectural styles. A noteworthy observation is that, in all cases, the analysis yields a high agreement between the obtained model-based performance predictions.

7 Concluding remarks

Engineering CASs constitutes a dynamic research field characterized by a multitude of challenges, with a primary focus on the adaptivity and autonomy of both agents and systems. This special issue comprehensively addresses a range of topics, including correctness by design and synthesis, bio-inspired communication systems, novel system models, machine learning, and the programming and analysis of ensembles. Notably, machine learning techniques play a crucial role in complementing rigorous (formal) methods, especially in data-driven applications and systems operating in dynamically changing environments and adapting to evolving requirements.

The story does not end here. In the near future, large language models such as OpenAI’s GPT [66], Google’s Bard

[68], and Meta's Llama [69] will offer new opportunities for software construction [10, 26] and may also influence the way how CASs are built. This might be the topic for the next special section of *STTT*.

Acknowledgements As the editors of this special section, we would like to extend our heartfelt thanks to all the authors for their valuable contributions and to all the reviewers for their careful evaluations and constructive comments. We are also grateful to the ISoLA chairs, Tiziana Margaria and Bernhard Steffen, for giving us the opportunity to organize this special section, and we thank them and *STTT* for providing us with the EquiNOCS system, which significantly facilitated the editorial process.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Abd Alrahman, Y., Piterman, N.: Modelling and verification of reconfigurable multi-agent systems. *Auton. Agents Multi-Agent Syst.* **35**(2), 47 (2021)
2. Abd Alrahman, Y., Azzopardi, S., Piterman, N.: Model checking reconfigurable interacting systems. In: [58], pp. 373–389 (2022)
3. Abd Alrahman, Y., Azzopardi, S., Stefano, L.D., Piterman, N.: Language support for verifying reconfigurable interacting systems. *Int. J. Softw. Tools Technol. Transf.* This issue. (2023)
4. Alrahman, Y.A., De Nicola, R., Loreti, M.: A calculus for collective-adaptive systems and its behavioural theory. *Inf. Comput.* **268** (2019)
5. Andrade, L.F., Baldan, P., Baumeister, H., Bruni, R., Corradini, A., De Nicola, R., Fiadeiro, J.L., Gadducci, F., Gnesi, S., et al.: AG-ILE: software architecture for mobility. In: Wirsing, M., Pattinson, D., Hennicker, R. (eds.) *WADT 2002, Revised Selected Papers*. *Lecture Notes in Computer Science*, vol. 2755, pp. 1–33. Springer, Berlin (2002)
6. Basile, D., ter Beek, M.H.: Contract automata library. *Sci. Comput. Program.* **221**, 102841 (2022)
7. Basile, D., ter Beek, M.H., Ciancia, V.: An experimental toolchain for strategy synthesis with spatial properties. In: [58], pp. 142–164 (2022)
8. Belmonte, G., Ciancia, V., Latella, D., Voxlogica, M.M.: A spatial model checker for declarative image analysis. In: Vojnar, T., Zhang, L. (eds.) *TACAS 2019, Proceedings, Part I*. *Lecture Notes in Computer Science*, vol. 11427, pp. 281–298. Springer, Berlin (2019)
9. Belzner, L., Hölzl, M.M., Koch, N., Wirsing, M.: Collective autonomic systems: towards engineering principles and their foundations. *LNCSTrans. Found. Mastering Chang.* **1**, 180–200 (2016)
10. Belzner, L., Gabor, T., Wirsing, M.: Large language model assisted software engineering: prospects, challenges, and a case study. In: *AISoLA 2023. Lecture Notes in Computer Science*, vol. 14380. Springer, Berlin (2023)
11. Bettini, L., De Nicola, R., Pugliese, R., Ferrari, G.: Interactive mobile agents in X-Klaim. In: *7th Workshop on Enabling Technologies (WETICE'98)*, pp. 110–117. *IEEE Comput. Soc., Los Alamitos* (1998)
12. Bettini, L., Nicola, R.D., Falassi, D., Lacoste, M., Lopes, L.M.B., Oliveira, L., Paulino, H., Vasconcelos, V.T.: A software framework for rapid prototyping of run-time systems for mobile calculi. In: Priami, C., Quaglia, P. (eds.) [64], pp. 179–207 (2004)
13. Bettini, L., Merelli, E., Tiezzi, F.: X-Klaim is back. In: Boreale, M., Corradini, F., Loreti, M., Pugliese, R. (eds.) *Models, Languages, and Tools for Concurrent and Distributed Programming - Essays Dedicated to Rocco De Nicola on the Occasion of His 65th Birthday*. *Lecture Notes in Computer Science*, vol. 11665, pp. 115–135. Springer, Berlin (2019)
14. Bettini, L., Bourr, K., Pugliese, R., Tiezzi, F.: Writing robotics applications with X-Klaim. In: [57], pp. 361–379 (2020)
15. Bettini, L., Bourr, K., Pugliese, R., Tiezzi, F.: Programming multi-robot systems with X-KLAIM. In: [58], pp. 283–300 (2022)
16. Bettini, L., Bourr, K., Pugliese, R., Tiezzi, F.: Coordinating and programming multiple ROS-based robots with X-KLAIM. *Int. J. Softw. Tools Technol. Transf.* This issue. (2023)
17. Bialek, W., Cavagna, A., Giardina, I., Mora, T., Silvestri, E., Viale, M., Walczak, A.M.: Statistical mechanics for natural flocks of birds. *Proc. Natl. Acad. Sci.* **109**(13), 4786–4791 (2012)
18. Bortolussi, L., De Nicola, R., Gast, N., Gilmoro, S., Hillston, J., Massink, M., Tribastone, M.: A quantitative approach to the design and analysis of collective adaptive systems. In: *1st FoCAS Workshop on Fundamentals of Collective Adaptive Systems* (2013)
19. Bortolussi, L., Hillston, J., Latella, D., Massink, M.: Continuous approximation of collective system behaviour: a tutorial. *Perform. Eval.* **70**(5), 317–349 (2013)
20. Boudec, J.L., McDonald, D.D., Munding, J.: A generic mean field convergence result for systems of interacting objects. In: *Fourth International Conference on the Quantitative Evaluation of Systems (QEST 2007)*, 17–19 September 2007, pp. 3–18. *IEEE Computer Society, Edinburgh, Scotland, UK* (2007)
21. Bozga, M., Sifakis, J.: Correct by design coordination of autonomous driving systems. In: [58], pp. 13–29 (2022)
22. Bozga, M., Sifakis, J.: Correct by design coordination of autonomous driving systems. *Int. J. Softw. Tools Technol. Transf.* This issue. (2023)
23. Broy, M.: A model of dynamic systems. In: Bensalem, S., Lakhnech, Y., Legay, A. (eds.) *From Programs to Systems. The Systems Perspective in Computing - ETAPS Workshop, FPS 2014, in Honor of Joseph Sifakis*. *Proceedings, Grenoble, France, April 6, 2014. Lecture Notes in Computer Science*, vol. 8415, pp. 39–53. Springer, Berlin (2014)
24. Bucchiarone, A., Dulay, N., Lavygina, A., Marconi, A., Raik, H., Russo, A.: An approach for collective adaptation in socio-technical systems. *SASO Workshops*, 43–48 (2015)
25. Bures, T., De Nicola, R., Gerostathopoulos, I., Hoch, N., Kit, M., Koch, N., Monreale, G.V., Montanari, U., Pugliese, R., Serbedzija, N.B., Wirsing, M., Zambonelli, F.: A life cycle for the development of autonomic systems: the e-mobility showcase. In: *7th IEEE International Conference on Self-Adaptation and Self-Organizing Systems Workshops, SASOW, 2013, Philadelphia, PA, USA, September 9–13, 2013*, pp. 71–76. *IEEE Comput. Soc., Los Alamitos* (2013)
26. Busch, D., Nolte, G., Bainsczyk, A., Steffen, B.: ChatGPT in the loop. In: *AISoLA 2023. Lecture Notes in Computer Science*, vol. 14380. Springer, Berlin (2023)

27. Cardelli, L., Tribastone, M., Tschaikowski, M.: From electric circuits to chemical networks. *Nat. Comput.* **19**(1), 237–248 (2020)
28. Casalaro, G.L., Cattivera, G., Ciccozzi, F., Malavolta, I., Wortmann, A., Pelliccione, P.: Model-driven engineering for mobile robotic systems: a systematic mapping study. *Softw. Syst. Model.* **21**(1), 19–49 (2022)
29. Chen, Y., Sanders, J.W.: A modal approach to consciousness of agents. In: [58], pp. 127–141 (2022)
30. Chen, Y., Sanders, J.W.: A modal approach to conscious social agents. *Int. J. Softw. Tools Technol. Transf.* This issue. (2023)
31. David, R., Alla, H.: Petri nets for modeling of dynamic systems: a survey. *Automatica* **30**(2), 175–202 (1994)
32. De Nicola, R., Ferrari, G., Pugliese, R.: KLAIM: a kernel language for agents interaction and mobility. *IEEE Trans. Softw. Eng.* **24**(5), 315–330 (1998)
33. De Nicola, R., Loreti, M., Pugliese, R., Tiezzi, F.: A formal approach to autonomic systems programming: the SCEL language. *ACM Trans. Auton. Adapt. Syst.* **9**(2), 7:1–7:29 (2014)
34. De Nicola, R., Jähnichen, S., Wirsing, M.: Rigorous engineering of collective adaptive ensembles – track introduction. In: [56], pp. 3–12 (2018)
35. De Nicola, R., Jähnichen, S., Wirsing, M.: Rigorous engineering of collective adaptive systems: special section. *Int. J. Softw. Tools Technol. Transf.* **22**(4), 389–397 (2020)
36. De Nicola, R., Maggi, A., Sifakis, J.: The dream framework for dynamic reconfigurable architecture modelling: theory and applications. *Int. J. Softw. Tools Technol. Transf.* **22**(4), 437–455 (2020)
37. De Nicola, R., Stefano, L.D., Inverso, O., Valiani, S.: Modelling flocks of birds from the bottom up. In: [58], pp. 82–96 (2022)
38. De Nicola, R., Stefano, L.D., Inverso, O., Valiani, S.: Modelling flocks of birds and colonies of ants from the bottom up. *Int. J. Softw. Tools Technol. Transf.* This issue. (2023)
39. Doerr, A., Daniel, C., Schiegg, M., Nguyen-Tuong, D., Schaal, S., Toussaint, M., Trimpe, S.: Probabilistic recurrent state-space models. In: Dy, J.G., Krause, A. (eds.) *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Proceedings of Machine Learning Research*, vol. 80, pp. 1279–1288. PMLR (2018)
40. Dorigo, M., Stützle, T.: *Ant Colony Optimization*. MIT Press, Cambridge (2004)
41. Fettke, P., Reisig, W.: Handbook of heraklit. <https://dblp.dagstuhl.de/db/conf/fmas/index.html>. Heraklit working paper, v1.1, September 20, 2021
42. Fettke, P., Reisig, W.: Discrete models of continuous behavior of collective adaptive systems. In: Margaria, T., Steffen, B. (eds.) [58], pp. 65–81 (2022)
43. Fettke, P., Reisig, W.: A causal, time independent synchronization pattern for collective adaptive systems. *Int. J. Softw. Tools Technol. Transf.* This issue. (2023)
44. Gheibi, O., Weyns, D., Quin, F.: Applying machine learning in self-adaptive systems: a systematic literature review. *ACM Trans. Auton. Adapt. Syst.* **15**(3), 9:1–9:37 (2020)
45. Gulwani, S., Polozov, O., Singh, R.: Program synthesis. *Found. Trends Program. Lang.* **4**(1–2), 1–119 (2017)
46. Hölzl, M.M., Rauschmayer, A., Wirsing, M.: Engineering of software-intensive systems: state of the art and research challenges. In Wirsing et al. [71], pp. 1–44
47. Hölzl, M.M., Wirsing, M.: Towards a system model for ensembles. In: *Formal Modeling: Actors, Open Systems, Biological Systems*. Lecture Notes in Computer Science, vol. 7000, pp. 241–261. Springer, Berlin (2011)
48. Horn, P.J.: *Autonomic Computing: IBM’s Perspective on the State of Information Technology*. IBM (2001)
49. Inverso, O., Trubiani, C., Tuosto, E.: Abstractions for collective adaptive systems. In: [57], pp. 243–260 (2020)
50. Jähnichen, S., Wirsing, M.: Rigorous engineering of collective adaptive systems – track introduction. In: [55], pp. 535–538 (2016)
51. Kernbach, S., Schmickl, T., Timmis, J.: Collective adaptive systems: challenges beyond evolvability (2011). [arXiv:1108.5643](https://arxiv.org/abs/1108.5643). CoRR
52. Macenski, S., Foote, T., Gerkey, B., Lalancette, C., Woodall, W.: *Robot Operating System 2: Design, architecture, and uses in the wild*. *Science Robotics* **7**(66) (2022)
53. Mahfoudh, H.B., Serugendo, G.D.M., Naja, N., Abdennadher, N.: Learning-based coordination model for spontaneous self-composition of reliable services in a distributed system. *Int. J. Softw. Tools Technol. Transf.* **22**(4), 417–436 (2020)
54. Margaria, T., Steffen, B. (eds.): *Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change - 6th International Symposium, ISoLA 2014, Part I*. Lecture Notes in Computer Science., vol. 8802. Springer, Berlin (2014)
55. Margaria, T., Steffen, B. (eds.): *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques - 7th International Symposium, ISoLA 2016, Part I*. Lecture Notes in Computer Science., vol. 9952. Springer, Berlin (2016)
56. Margaria, T., Steffen, B. (eds.): *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques - 8th International Symposium, ISoLA 2018, Part III*. Lecture Notes in Computer Science., vol. 11246. Springer, Berlin (2018)
57. Margaria, T., Steffen, B. (eds.): *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques - 9th International Symposium, ISoLA 2020, Part II*. Lecture Notes in Computer Science., vol. 12477. Springer, Berlin (2020)
58. Margaria, T., Steffen, B. (eds.): *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques - 8th International Symposium, ISoLA 2022, Part III*. Lecture Notes in Computer Science., vol. 13703. Springer, Berlin (2022)
59. Monica, S., Bergenti, F., Zambonelli, F.: Toward a kinetic framework to model the collective dynamics of multi-agent systems. In: [58], pp. 165–180 (2022)
60. Monica, S., Bergenti, F., Zambonelli, F.: A kinetic approach to study the collective dynamics of multi-agent systems. *Int. J. Softw. Tools Technol. Transf.* This issue. (2023)
61. Murgia, M., Pinciroli, R., Trubiani, C., Tuosto, E.: Comparing performance abstractions for collective adaptive systems. *Int. J. Softw. Tools Technol. Transf.* This issue. (2023)
62. Olfati-Saber, R.: Flocking for multi-agent dynamic systems: algorithms and theory. *IEEE Trans. Autom. Control* **51**(3), 401–420 (2006)
63. Pinciroli, R., Trubiani, C.: Model-based performance analysis for architecting cyber-physical dynamic spaces. In: *18th IEEE International Conference on Software Architecture, ICSA 2021, Stuttgart, Germany, March 22–26, 2021*, pp. 104–114. IEEE (2021)
64. Priami, C., Quaglia, P. (eds.): *Global Computing, IST/FET International Workshop, GC 2004, Rovereto, Italy, March 9–12, 2004*. Lecture Notes in Computer Science, vol. 3267. Springer, Berlin (2005)
65. Puviani, M., Cabri, G., Zambonelli, F.: A taxonomy of architectural patterns for self-adaptive systems. In: Desai, B.C., Almeida, A.M., Bernardino, J., Mudur, S.P. (eds.) *International C* Conference on Computer Science & Software Engineering, C3S2E13, Porto, Portugal, July 10–12, 2013*, pp. 77–85. ACM, New York (2013)
66. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners. OpenAI, San Francisco, California, United States, 2019. https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf, accessed on July 05, 2023
67. Basile, D., ter Beek, M., Bussi, L., Ciancia, V.: A toolchain for strategy synthesis with spatial properties. *Int. J. Softw. Tools Technol. Transf.* This issue. (2023)

68. Thoppilan, R., De Freitas, D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H.-T., Jin, A., Bos, T., Baker, L., Du, Y., et al.: *Lambda: Language models for dialog applications* (2022). ArXiv preprint. [arXiv:2201.08239](https://arxiv.org/abs/2201.08239)
69. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., Llama, G.L.: *Open and efficient foundation language models* (2023). [arXiv:2302.13971](https://arxiv.org/abs/2302.13971). CoRR
70. Wirsing, M., Belzner, L.: *Towards systematically engineering autonomous systems using reinforcement learning and planning*. In: López-García, P., Gallagher, J.P., Giacobazzi, R. (eds.) *Analysis, Verification and Transformation for Declarative Programming and Intelligent Systems - Essays Dedicated to Manuel Hermenegildo on the Occasion of His 60th Birthday*. Lecture Notes in Computer Science, vol. 13160, pp. 281–306. Springer, Berlin (2023)
71. Wirsing, M., Banâtre, J.-P., Hölzl, M.M., Rauschmayer, A. (eds.): *Software-Intensive Systems and New Computing Paradigms - Challenges and Visions*. Lecture Notes in Computer Science., vol. 5380. Springer, Berlin (2008)
72. Wirsing, M., Hölzl, M.M., Tribastone, M., Zambonelli, F.: *ASCENS: engineering autonomic service-component ensembles*. In: Beckert, B., Damiani, F., de Boer, F.S., Bonsangue, M.M. (eds.) *Formal Methods for Components and Objects, 10th International Symposium, FMCO 2011*. Lecture Notes in Computer Science, vol. 7542, pp. 1–24. Springer, Turin, Italy (2011). October 3-5, 2011, Revised Selected Papers
73. Wirsing, M., De Nicola, R., Hölzl, M.M.: *Rigorous engineering of autonomic ensembles – track introduction*. In: [54], pp. 96–98 (2014)
74. Wirsing, M., Hölzl, M.M., Koch, N., Mayer, P. (eds.): *Software Engineering for Collective Autonomic Systems – the ASCENS Approach*. Lecture Notes in Computer Science., vol. 8998. Springer, Berlin (2015)
75. Wirsing, M., De Nicola, R., Jähnichen, S.: *Rigorous engineering of collective adaptive systems introduction to the 3rd track edition*. In: Margaria, T., Steffen, B. (eds.) [57], pp. 161–170 (2020)
76. Wirsing, M., De Nicola, R., Jähnichen, S.: *Rigorous engineering of collective adaptive systems introduction to the 4th track edition*. In: [58], pp. 3–12 (2022)
77. Zambonelli, F., Castelli, G., Ferrari, L., Mamei, M., Rosi, A., Di Marzo Serugendo, G., Risoldi, M., Tchao, A.-E., Dobson, S., Stevenson, G., Ye, J., Nardini, E., Omicini, A., Montagna, S., Viroli, M.: *Self-aware pervasive service ecosystems*. In: *The Europ. Future Technologies Conf. and Exhibition, Procedia Computer Science 7*, pp. 197–199 (2011)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.